POLITECNICO
MILANO 1863

Computer Science and Engineering

PowerEnjoy Service - Integration Test Plan Document

December 31, 2016

Prof. Luca Mottola

Authors:

- ZHOU YINAN(Mat. 872686)
- ZHAO KAIXIN(Mat. 875464)
- ZHAN YUAN(Mat. 806508)

Contents

1 INTRODUCTION **2**
- 1.1 Revision History 2
- 1.2 Purpose and Scope 2
- 1.3 List of Definitions and Abbreviations 2
- 1.4 List of Reference Documents 2

2 INTEGRATION STRATEGY **3**
- 2.1 Entry Criteria 3
- 2.2 Elements to be Integrated 3
- 2.3 Integration Testing Strategy 4
- 2.4 Sequence of Component/Function Integration 4
  - 2.4.1 Software Integration Sequence 4
  - 2.4.2 Subsystem Integration Sequence 7

3 Individual Steps and Test Description **8**

4 Tools and Test Equipment Required **9**

5 Program Stubs and Test Data Required **10**

6 Effort Spent **11**

1 INTRODUCTION

1.1 Revision History

At this moment, this is the first version of the document.

1.2 Purpose and Scope

This document describes how the integration should proceed. Integration testing means that we need to verify all the components needed for the overall system should work correctly not only individually but also in combination. In this document, we provide the steps needed to follow in order to get a fully functional system. More specifically, the elements need to be tested, the testing strategy, sequence of integration, test description, tools and stubs will be presented in the following parts.

1.3 List of Definitions and Abbreviations

- RASD : Requirement Analysis and Specification Document
- DD : Design Document
- Guest : All the users of the system who have not performed a Log in operation yet
- User : After a Guest logs in, he/she becomes a User
- Subcomponent : each of the low level component realizing specific functionalities of the subsystem
- subsystem : a functional unit of the system

1.4 List of Reference Documents

- Assignment AA 2016-2017
- RASD
- DD

2 INTEGRATION STRATEGY

2.1 Entry Criteria

There are several entry criteria to be completed before the integration testing phase can begin.

- RASD and DD documents are completed
- Components have to be unit tested before the integration testing
- The required driver and stub have already been developed
- database is fully functioned

The application subsystem may not be fully developed at this moment, however the interface between Application tier and Server tier is a must for testing to proceed.

2.2 Elements to be Integrated

The system is divided into 3 subsystems according to the 3 tier architecture we chose in the DD : Application, Server, Database. This document mainly focuses on the integration testing for the Server side. The following components needed to be integrated:

- Guest Application Manager
- User Application Manager
- Car Application Manager
- Database Manager

The above components are the basic low-level components required for higher level functionalities of the system. Besides the components we need to develop by ourselves, some external systems and API are used:

- Google Map API
- Bank Service system

2.3 Integration Testing Strategy

For testing the integration of components, we choose the bottom-up approach. By bottom-up approach, we start by the components which have no dependency of other components and the very fundamental components providing services to all others. In our system, we start from the Database Manager component. The reason behind it is that basically all of our functions need Database Manager. Thus it is natural and easy to begin with it (the bottom level) and add other components step by step.

2.4 Sequence of Component/Function Integration

Basically we have three subsystems in our system : Application side, Server side and Database side. We will focus on the first two subsystem and neglect the last one because we'll use a DBMS from outside.

2.4.1 Software Integration Sequence

(1) Application side The application subsystem is composed of two components : User application and Car application. These two components are parallel and do not have any dependency on each other. The dependency of these two components lie on the Server side. Thus, these two components can be developed separately but can only be tested after the Server side is functional.

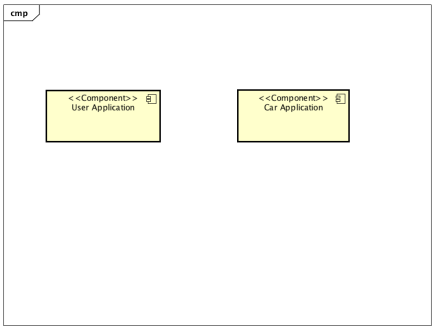

Figure 1: figure Application subsystem

(2) Server side The Server has four components : Database Manager, Guest Application Manager, User Application Manager and Car Application Manager. We'll start from Database Manager and add other components step by step.

- 1 Database Manager

Database Manager is the most basic and fundamental component in our system, thus it will be tested firstly.

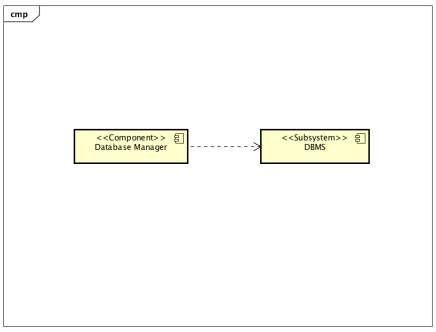

Figure 2: Database Manager

- 2.1 Gust Application Manager

After testing the Database Manager, we'll proceed with our testing procedure with Guest Application Manager which is responsible for user registration and log in. Note the this testing can be proceeded in parallel with the Car Application Manager.

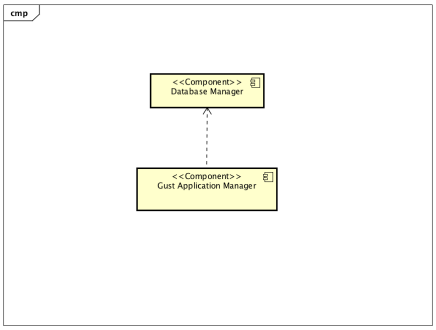

Figure 3: Guest Application Manager

- 2.2 Car Application Manager

Car Application Manager is responsible for managing the car information with database.

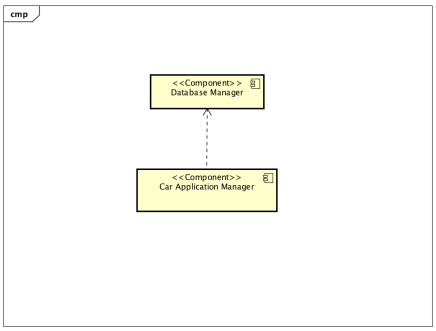

Figure 4: Car Application Manager

- 3 User Application Manager

All the services for the users can only be accessed after performing the log in operation. However User Application Manager and Guest Application Manager do not have dependency on each other. User Application Manager is responsible for rental services. More preciously, getting available cars, making reservations and unlocking the door.

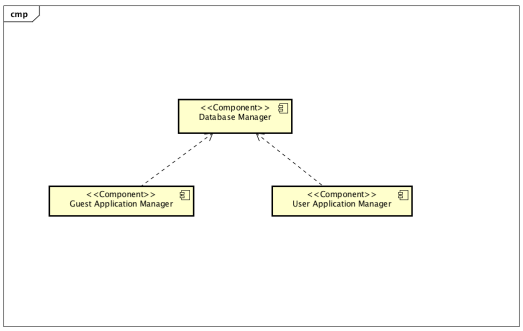

Figure 5: User Application Manager

2.4.2 Subsystem Integration Sequence

After testing the components in the Server and Application sides, we can begin to test the integration of subsystems. Since a full functional requirement needs to operate on all the three subsystems, we integrate them all together.

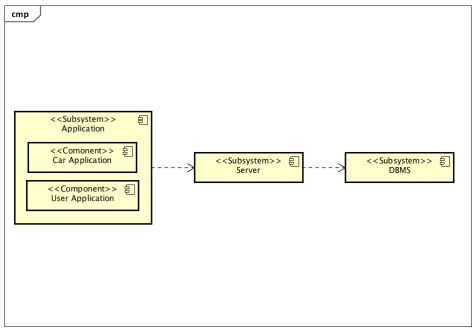

Figure 6: subsystems

Here presents the overall dependency graph for the whole system.

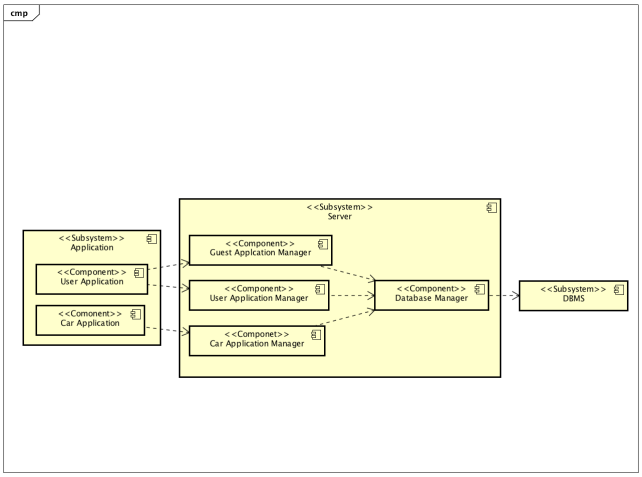

Figure 7: subsystems

3 Individual Steps and Test Description

4 Tools and Test Equipment Required

5 Program Stubs and Test Data Required

6 Effort Spent